

Figure 1

```
[SqlUserDefinedType (Format.Native)]  
public class Point  
{  
    float x;  
    float y;  
  
    public Bool IsNull {get; }  
    public static Point Null {get; }  
  
    public String ToString()  
    public Static Point Parse (SqlString s)  
}
```

Figure 2

```
0000: (04)    ldarg.2  
0001: (15)    ldc.i4.m1  
0002: (05)    ldarg.3  
0003: (28)    call        06000113 (Point, geometry, Version=0.0.0.0,  
Culture=neutral, PublicKeyToken=0f152244995ad2ef::Deserialize)  
0008: (04)    ldarg.2  
0009: (1F)    ldc.i4.s    24  
000B: (58)    add  
000C: (15)    ldc.i4.m1  
000D: (05)    ldarg.3  
000E: (28)    call        06000113 (Point, geometry, Version=0.0.0.0,  
Culture=neutral, PublicKeyToken=0f152244995ad2ef::Deserialize)  
0013: (6F)    callvirt    0A00008E (Distance)  
0018: (0A)    stloc.0  
0019: (03)    ldarg.1  
001A: (06)    ldloc.0  
001B: (28)    call        0A00003B (CallData_Int32ToI4)  
0020: (DD)    leave      0025  
0025: (2A)    ret  
0026: (28)    call        0A000004 (CallData_CopyCLRExceptionInfoToNonManaged)  
002B: (DD)    leave      0030  
0030: (2A)    ret
```

Figure 3

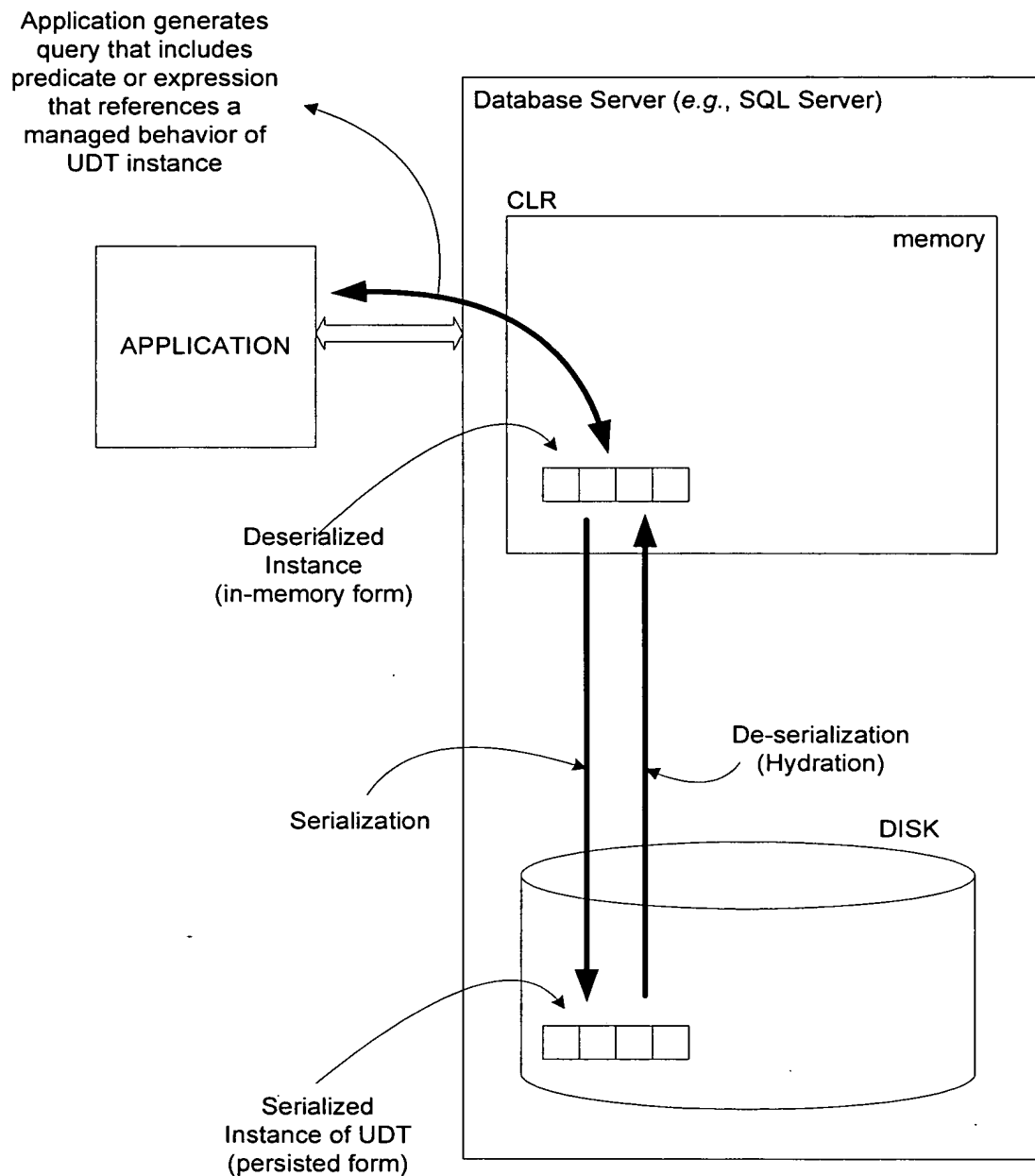


Figure 4

```

0000: (04)    ldarg.2
0001: (15)    ldc.i4.m1
0002: (05)    ldarg.3
0003: (28)    call      06000113 (Point, geometry, Version=0.0.0.0,
Culture=neutral, PublicKeyToken=0f152244995ad2ef::Deserialize)
0008: (0A)    stloc.0
0009: (06)    ldloc.0
000A: (04)    ldarg.2
000B: (1F)    ldc.i4.s    24
000D: (58)    add
000E: (46)    ldind.i1
000F: (39)    brfalse    002D
0014: (20)    ldc.i4      6569
0019: (17)    ldc.i4.1
001A: (1F)    ldc.i4.s    20
001C: (18)    ldc.i4.2
001D: (20)    ldc.i4      1133834472
0022: (28)    call      0600000F (Method_RaiseSpecificError)
0027: (73)    newobj     0A000003 (Ctor_ControlFlowException)
002C: (7A)    throw
002D: (04)    ldarg.2
002E: (1F)    ldc.i4.s    24
0030: (58)    add
0031: (28)    call      0A000039 (CallData_I4ToInt32)
0036: (6F)    callvirt   0A00008D (set_X)
003B: (03)    ldarg.1
003C: (20)    ldc.i4      61443
0041: (54)    stind.i4
0042: (06)    ldloc.0
0043: (39)    brfalse    0053
0048: (06)    ldloc.0
0049: (6F)    callvirt   0A00008C (Point, geometry, Version=0.0.0.0,
Culture=neutral, PublicKeyToken=0f152244995ad2ef::IsNull)
004E: (39)    brfalse    005F
0053: (03)    ldarg.1
0054: (20)    ldc.i4      61443
0059: (54)    stind.i4
005A: (38)    br         00CA
005F: (20)    ldc.i4      8000
0064: (8D)    newarr     0100002E (Type_System_Byte)
0069: (73)    newobj     0A000011 (Ctor_SqlBytes_With_ByteArray)
006E: (0B)    stloc.1
006F: (06)    ldloc.0
0070: (74)    castclass   01000032 (Interface_IBinarySerialize)
0075: (07)    ldloc.1
0076: (28)    call      0A000010 (CallData_SqlBytes_GetStream)
007B: (0C)    stloc.2
007C: (08)    ldloc.2
007D: (73)    newobj     0A000006 (Ctor_BinaryWriter)
0082: (28)    call      0A000088 (Point, geometry, Version=0.0.0.0,
Culture=neutral, PublicKeyToken=0f152244995ad2ef::Write

```

Figure 5A

```

0087: (14)    ldnull
0088: (0D)    stloc.3
0089: (07)    ldloc.1
008A: (28)    call        0A000015 (PropGet_SqlBytes_IsNull)
008F: (39)    brfalse    009C
0094: (16)    ldc.i4.0
0095: (13)    stloc.s    4
0097: (38)    br        00AC
009C: (07)    ldloc.1
009D: (28)    call        0A000013 (PropGet_SqlBytes_Buffer)
00A2: (0D)    stloc.3
00A3: (08)    ldloc.2
00A4: (6F)    callvirt    0A000019 (CallData_Stream_Prop_Position)
00A9: (69)    conv.i4
00AA: (13)    stloc.s    4
00AC: (09)    ldloc.3
00AD: (11)    ldloc.s    4
00AF: (03)    ldarg.1
00B0: (20)    ldc.i4      8000
00B5: (17)    ldc.i4.1
00B6: (17)    ldc.i4.1
00B7: (20)    ldc.i4      257
00BC: (16)    ldc.i4.0
00BD: (28)    call        0A000009 (CallData_XVariantFromSqlBytesBuffer)
00C2: (26)    pop
00C3: (03)    ldarg.1
00C4: (20)    ldc.i4      61440
00C9: (54)    stind.i4
00CA: (DD)    leave      00CF
00CF: (2A)    ret
00D0: (28)    call        0A000004
(CallData_CopyCLRExceptionInfoToNonManaged)
00D5: (DD)    leave      00DA
00DA: (2A)    ret

```

Figure 5B

Column name	Data Type	Description
name	sysname	Name of type, unique within the schema.
user_type_id	int	Id of type, unique within the database.
schema_id	int	Id of schema to which type belongs.
principal_id	int	Id of individual owner if different from schema owner.
system_type_id	tinyint	Id of type's internal system-type.
max_length	smallint	Maximum length of type in bytes.
precision	tinyint	Maximum precision of type if numeric-based, else 0.
scale	tinyint	Maximum scale of type if numeric-based, else 0.
collation_name	sysname	Name of type's collation if character-based, else NULL.
is_nullable	bit	1 if type is nullable, else 0.
is_user_defined	bit	1 if this is a user-defined type. 0 if this is a built-in type that ships with SQL Server.
is_assembly_type	bit	1 if this is a type whose implementation is defined in a CLR assembly. 0 if this type is based on native SQL Server types.
default_object_id	int	Id of the stand-alone default bound to the type via sys.sp_bindefault
rule_object_id	int	Id of the stand-alone rule bound to the type via sys.sp_bindrule

Figure 6

Column Name	Data Type	Description
assembly_id	int	Id of assembly from which this type was created.
assembly_class	sysname	Name of class within assembly which defines this type.
is_binary_ordered	bit	Sorting the bytes of this type is equivalent to sorting using comparison operators on type.
is_fixed_length	bit	Length of type is always same as max_length.
prog_id	nvarchar (40)	ProgID of the type as exposed to COM

Figure 7

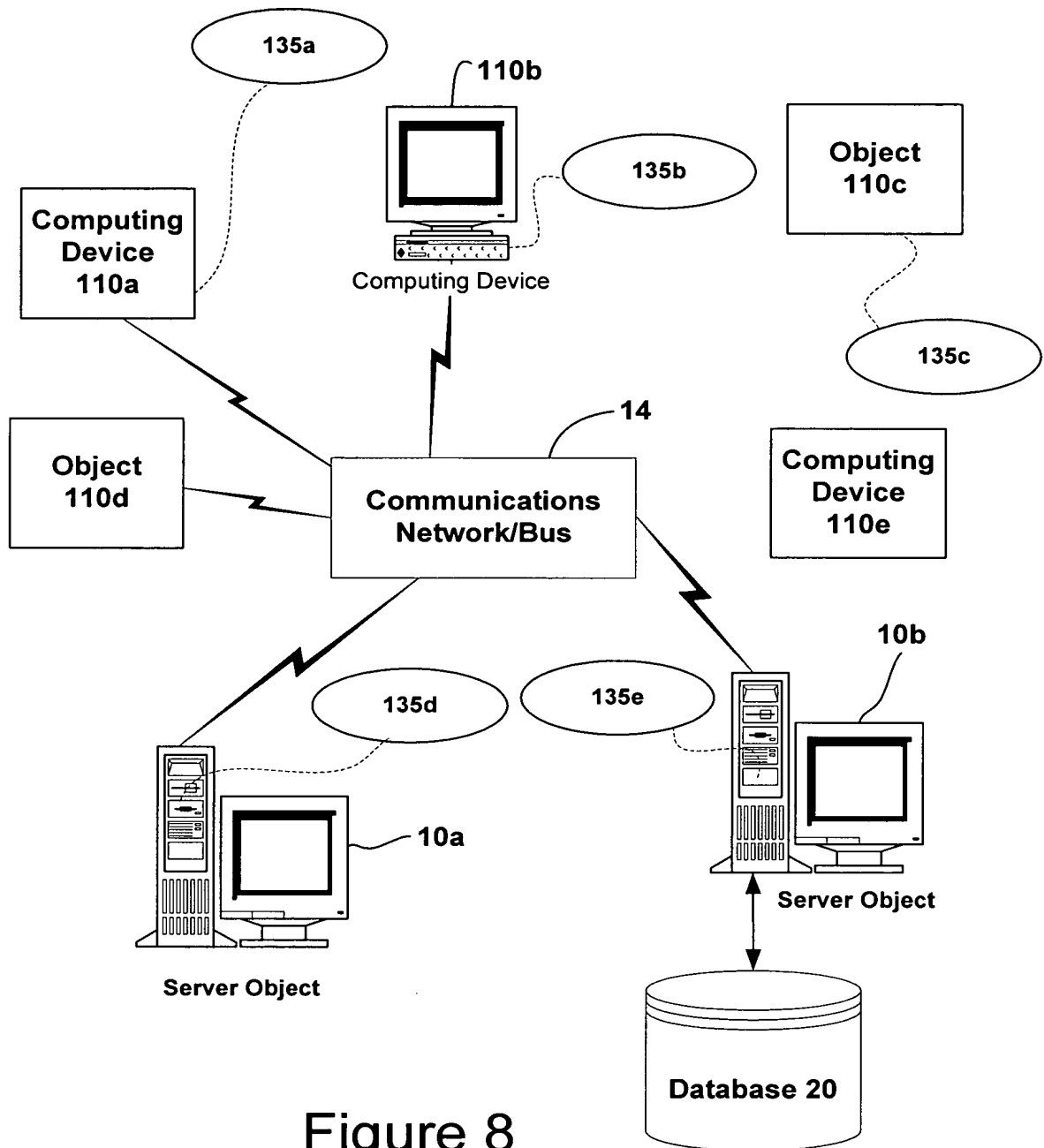


Figure 8

Computing Environment 100

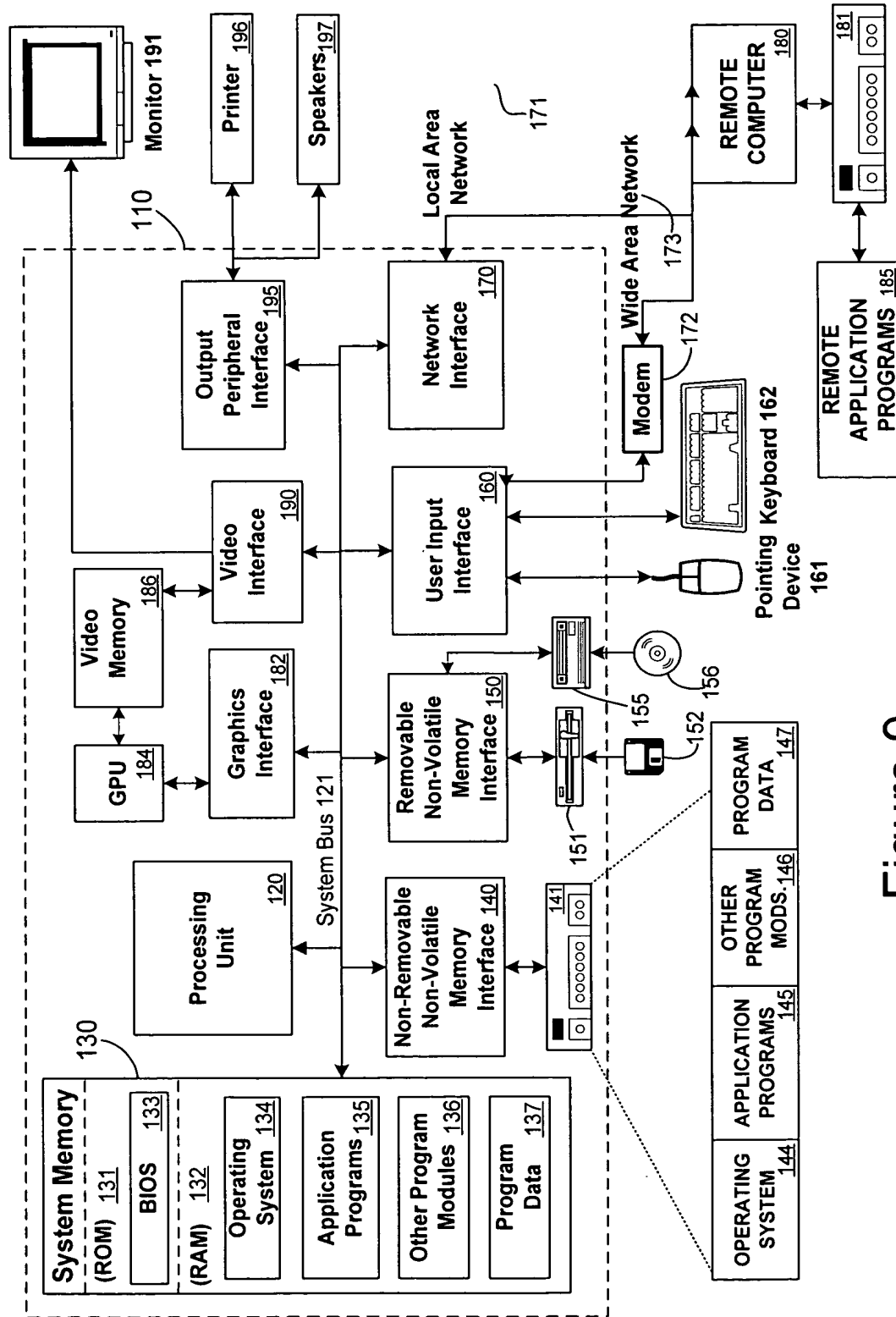


Figure 9